Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Linear Algebra and its Applications 415 (2006) 52–81

LINEAR ALGEBRA AND ITS APPLICATIONS

www.elsevier.com/locate/laa

Krylov type subspace methods for matrix polynomials

Leonard Hoffnung [1], Ren-Cang Li [*,2], Qiang Ye [3]

Department of Mathematics, University of Kentucky, Lexington, KY 40506, USA

Accepted 16 September 2005

Available online 2 November 2005

Submitted by Z. Bai

Abstract

We consider solving eigenvalue problems or model reduction problems for a quadratic matrix polynomial $I\lambda^2 - A\lambda - B$ with large and sparse A and B. We propose new Arnoldi and Lanczos type processes which operate on the same space as A and B live and construct projections of A and B to produce a quadratic matrix polynomial with the coefficient matrices of much smaller size, which is used to approximate the original problem. We shall apply the new processes to solve eigenvalue problems and model reductions of a second order linear input–output system and discuss convergence properties. Our new processes are also extendable to cover a general matrix polynomial of any degree.

AMS classification: 65F15; 65F20; 15A18

Keywords: Quadratic matrix polynomial; Krylov subspace; Quadratic eigenvalue problem; Model reduction

* Corresponding author.

E-mail addresses: lhoff@ms.uky.edu (L. Hoffnung), rcli@ms.uky.edu (R.-C. Li), qye@ms.uky.edu (Q. Ye).

[1] Supported in part by the NSF grant nos. CCR-9875201 and CCR-0098133.

[2] Supported in part by NSF CAREER award grant no. CCR-9875201 and by NSF grant no. DMS-0510664.

[3] Supported in part by NSF grant no. CCR-0098133 and NSF grant no. DMS-0411502.

0024-3795/$ - see front matter © 2005 Elsevier Inc. All rights reserved.

doi:10.1016/j.laa.2005.09.016

1. Introduction

Krylov subspace techniques are widely used for solving linear systems of equations and eigenvalue problems involving large and sparse matrices [7,13]. They have found applications in many other large scale matrix problems such as model reductions of linear input–output systems [10,11]. The basic idea of the techniques is to extract information of an $n \times n$ matrix A most relevant to the underlying computational problem through utilizing the so-called *Krylov* subspace

$$\mathscr{K}_k(A, v) = \operatorname{span}\{v, Av, \ldots, A^{k-1}v\},$$

or through utilizing two (row and column) Krylov subspaces $\mathscr{K}_k(A, v)$ and $\mathscr{K}_k(A^*, w)$ simultaneously, where v and w are vectors of dimension n, the asterisk denotes the conjugate transpose. This is realized by either the *Arnoldi* (or *Lanczos*) *process* when only $\mathscr{K}_k(A, v)$ is employed or by the *nonsymmetric Lanczos process* if both $\mathscr{K}_k(A, v)$ and $\mathscr{K}_k(A^*, w)$ are used [1,17]. See also [7,13,20,24,25].

When $\mathscr{K}_k(A, v)$ has dimension k, the Arnoldi process generates an orthonormal basis $\{q_1, q_2, \ldots, q_k\}$ for $\mathscr{K}_k(A, v)$, and an upper Hessenberg matrix $H_k = Q_k^* A Q_k$, which is the projection of A onto $\mathscr{K}_k(A, v)$, where $Q_k = [q_1, q_2, \ldots, q_k]$. On the other hand, the Lanczos process generates a basis for $\mathscr{K}_k(A, v)$ and a basis for $\mathscr{K}_k(A^*, w)$ such that the two bases are biorthogonal. Simultaneously, a tridiagonal matrix T_k is obtained, which is the projection of A onto $\mathscr{K}_k(A, v)$ along $\mathscr{K}_k(A^*, w)$. For Hermitian A, usually w is taken to be v and the process coincides with the Arnoldi process and is called the (symmetric) Lanczos process.

For modest $k \ll n$, some of the eigenvalues of H_k (and T_k) are good approximations to some eigenvalues (usually extreme part) of A. This approximation of A by H_k (and T_k) can be used in many other applications as well, such as the model reduction of linear input–output systems [5,10,11]. Over the years, many technical inventions, including a shift-and-invert strategy, look-ahead techniques, rational Krylov, (adaptive) block versions, implicit restart strategies, and so on, have been developed for the Lanczos/Arnoldi algorithm for better numerical efficiency and stability. But we shall not discuss them here, see for example [3,4,9,14,15,21,23,24,27,31] and references therein.

In this paper, we consider a related problem for a large and sparse $n \times n$ *monic matrix polynomial*

$$A(\lambda) = I_n \lambda^m - \sum_{i=0}^{m-1} A_i \lambda^i, \tag{1.1}$$

where I_n is the $n \times n$ identity matrix (later we may simply write I if its dimension is clear from the context). Eq. (1.1) is typically associated with the initial value problem for

$$x^{(m)}(t) - \sum_{i=0}^{m-1} A_i x^{(i)}(t) = g(t), \tag{1.2}$$

where A_i is an $n \times n$ constant matrix and $x(t)$ is a vector of dimension n, depending on t, and the initial values are usually given to $x^{(i)}(t)$ at $t = 0$ for $0 \leqslant i \leqslant m-1$. A modal analysis of (1.2) is to find those scalars μ and nonzero vectors x and/or y such that

$$A(\mu)x = 0, \quad y^*A(\mu) = 0.$$

μ is called an eigenvalue of (1.1) and x (and y) a right (and left, resp.) eigenvector. On the other hand, in a linear input–output system with the state governed by (1.2), we are interested in approximations and computations of the transfer function

$$f(s) = c^*A(s)^{-1}b. \tag{1.3}$$

Specifically, we would like to find a lower dimensional linear input–output system whose input–output relation (i.e., the transfer function) gives a good approximation to those of the given system. This is often referred to as model reductions.

In most cases, a problem (e.g., eigenvalue problem) concerning the matrix polynomial can be reduced to one for the following $nm \times nm$ matrix [12]:

$$A_{\mathrm{LIN}} = \begin{pmatrix} 0 & I & 0 & \cdots & 0 \\ 0 & 0 & I & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & I \\ A_0 & A_1 & A_2 & \cdots & A_{m-1} \end{pmatrix},$$

to which well-established methods can be applied. This is called *linearization*. For the eigenvalue problem or the model reduction problem, one can use the Arnoldi or the Lanczos algorithm on A_{LIN} to produce a Krylov subspace and then a projection, which is used to approximate A_{LIN}. Such a process will have to operate with vectors of dimension mn. Furthermore, the projection of A_{LIN} on a Krylov subspace is usually not a linearization of any matrix polynomial and thus the approximation as obtained loses its intrinsic physical connection to the original problem. In model reductions, for example, a consequence of this is that the reduced model that is obtained by applying the Arnoldi or the Lanczos process to the linearization problem A_{LIN} cannot be synthesized with a physical model of an mth order input–output system [2].

In this paper, we study extensions of the *standard Arnoldi process* and the *standard Lanczos process* for matrix polynomials without going through any linearization. We note that several other methods [19,26] have been developed that do not rely on the linearization processes (see also [4,28]). Here, we develop Krylov type projection methods that generate a basis $\{q_1, q_2, \ldots, q_k\}$ for a subspace as defined by A_i and its powers, and then apply projection simultaneously to each matrix A_i of the matrix polynomial to obtain $H_k^{(i)} = Q_k^* A_i Q_k$ which is also in some condensed form, where $Q_k = [q_1, q_2, \ldots, q_k]$. Then we approximate $A(\lambda)$ by the lower dimensional matrix polynomial

$$H_k(\lambda) \equiv I_k\lambda^m - \sum_{i=0}^{m-1} H_k^{(i)}\lambda^i.$$

Compared with the linearization, this approach has advantages of preserving certain properties of the original system $A(\lambda)$ such as

- If the field of values of $A(\lambda)$ (i.e., $\mathscr{F} = \{\lambda : x^*A(\lambda)x = 0 \text{ for some } x \neq 0\}$) is on the left half complex plane, which guarantees stability of the system (1.2), then the field of values of $H_k(\lambda)$ is also on the left half plane, preserving the stability.
- The property that A_i is symmetric, positive definite, etc. is preserved by $H_k^{(i)}$. In particular, if $A(\lambda)$ is an overdamped vibrating system [8,12], so is $H_k(\lambda)$. So the property that all eigenvalues are real is preserved. The same is true if $A(\lambda)$ comes from a weakly damped system, which has the eigenvalues near the imaginary axis.
- A gyroscopic system about a stable equilibrium [8] has $A_0 > 0$ and A_1 skew-Hermitian with $m = 2$. In this case, the eigenvalues are all imaginary. Again, the projection problem preserves this property.

An Arnoldi process of this type has already been developed recently for a monic quadratic polynomial [18], where a special case is also considered in which a linear combination of the two matrices is of low rank. The present paper, as a continuation of [18], will first investigate the subspaces associated with the Arnoldi type process, especially for the commutative case. This is done in Section 2. In Section 3, we develop a Lanczos type process for monic quadratic polynomials. Briefly analogous extensions to a general monic polynomial of degree m are mentioned. The uses of the Arnoldi/Lanczos type processes for model reductions and eigenvalue computations, and their convergent properties are given in Sections 4 and 5. We shall also present a numerical example to illustrate our new algorithms in Section 6. Throughout the paper, however, we will be focusing mostly on explaining the new ideas and leave subtle but important numerical considerations and numerical applications (e.g., more extensive numerical testing) to future studies. We also note that our focus on monic matrix polynomials does not lose any generality because non-monic cases can be handled implicitly as a monic one through some combination of shifting and a factorization of the leading matrix.

Notation. The jth column of an identity matrix (of size that will be clear from the context) is denoted as e_j. $\mathbb{C}^{m\times n}$ is the set of all m-by-n complex matrices, $\mathbb{C}^n = \mathbb{C}^{n\times 1}$ (vectors), and $\mathbb{C} = \mathbb{C}^1$ (scalars). We shall use MATLAB-like notation $X_{(i:j,k:\ell)}$ to denote the submatrix of X, consisting of the intersections of rows i to j and columns k to ℓ, and when $i : j$ is replaced by :, it means all rows, similarly for columns. The generic notation $\mathtt{x}$ is for a possible nonzero scalar, vector, and $\mathtt{X}$ for a possible nonzero matrix. $\|x\|_2$ is the Euclidean norm of a vector x and $\|X\|_2$ is the spectral norm of a matrix X. Given a real number γ, $\lfloor\gamma\rfloor$ is the largest integer that is not greater than γ.

2. Arnoldi type process for monic matrix polynomials

In this section, we shall first describe the Arnoldi type process for $I\lambda^2 - A\lambda - B$ derived in [18], and then study the Krylov type subspace that it computes. In particular, we discuss the special case when A and B commute. We shall also show how this can be generalized to a general mth degree monic matrix polynomial.

2.1. Arnoldi type process for $I\lambda^2 - A\lambda - B$

The Arnoldi type algorithm for $I\lambda^2 - A\lambda - B$ of [18] is based on the fact that given $q_1 \in \mathbb{C}^n$ with $\|q_1\|_2 = 1$, there is a unitary matrix $Q \in \mathbb{C}^{n\times n}$ with $Qe_1 = q_1$ such that[4]

$$Q^*AQ = H_a \equiv (h_{a;ij}), \quad Q^*BQ = H_b \equiv (h_{b;ij}) \tag{2.1}$$

with

$$h_{a;ij} = 0 \text{ for } i \geqslant 2j+1, \quad h_{b;ij} = 0 \text{ for } i \geqslant 2j+2. \tag{2.2}$$

From this, the following algorithm is derived in [18].

Algorithm 2.1 (Arnoldi Type Process)

1. **Given** q_1 **with** $\|q_1\|_2 = 1$;
2. $N = 1$;
3. **for** $j = 1, 2, \ldots, k$ **do**
4. **if** $j > N$ **then BREAK**;
5. $\hat{q} = Aq_j$;
6. **for** $i = 1, 2, \ldots, N$ **do**
7. $h_{a;ij} = q_i^*\hat{q}$; $\hat{q} = \hat{q} - q_i h_{a;ij}$;
8. **end for**
9. $h_{a;N+1,j} = \|\hat{q}\|_2$;
10. **if** $h_{a;N+1,j} > 0$ **then**
11. $N = N + 1$, $q_N = \hat{q}/h_{a;Nj}$;
12. **end if**
13. $\hat{q} = Bq_j$;
14. **for** $i = 1, 2, \ldots, N$ **do**
15. $h_{b;ij} = q_i^*\hat{q}$; $\hat{q} = \hat{q} - q_i h_{b;ij}$;
16. **end for**
17. $h_{b;N+1,j} = \|\hat{q}\|_2$;
18. **if** $h_{b;N+1,j} > 0$ **then**

[4] The entries of H_a and H_b are unfortunately heavily subscripted – with a lower case letter before the separator ";" to indicate the association to A or B, and with two integers i and j as their row and column indexes. For better readability, sometimes we insert a comma between the two integers.

19. $N = N + 1$; $q_N = \hat{q}/h_{a;Nj}$;
20. **end if**
21. **end for**

In the algorithm, N tracks the number of vectors q_i already generated at any given point. Let α_k and β_k be the values of N at the ends of Lines 12 and 20, respectively, for the trip $j = k$. It can be seen that $\alpha_k \leqslant \beta_k \leqslant \alpha_k + 1$. Upon completion of the above process, we have (see [18])

$$AQ_{(:,1:k)} = Q_{(:,1:\alpha_k)} H_{a(1:\alpha_k,1:k)}, \tag{2.3}$$

$$BQ_{(:,1:k)} = Q_{(:,1:\beta_k)} H_{b(1:\beta_k,1:k)}, \tag{2.4}$$

unless the j-loop is forced to **BREAK** out at Line 4, in which case,

$$AQ_{(:,1:N)} = Q_{(:,1:N)} H_{a(1:N,1:N)}, \tag{2.5}$$

$$BQ_{(:,1:N)} = Q_{(:,1:N)} H_{b(1:N,1:N)}. \tag{2.6}$$

While H_a and H_b are lower banded, their lower bandwidths increase quickly. In [18], the special case that a linear combination of A and B is of low rank is considered. In that case, the lower bandwidth of H_a and H_b is bounded by the rank of the combination plus 1. Later, we shall consider a special case when A and B commute. As we shall see, the lower bandwidth can also be significantly reduced in this case. We first need to describe the subspace $\text{span}\{q_1, q_2, \ldots, q_\ell\}$ in terms of q_1, A, and B.

2.2. *Subspace* $\text{span}\{q_1, q_2, \ldots, q_\ell\}$

Suppose first that in line 10 and 18 of Algorithm 2.1, all

$$h_{a;N+1,j} > 0, \quad h_{b;N+1,j} > 0.$$

We notice that the process starts with q_1, and at $j = 1$ it generates new directions in vectors Aq_1 first and then Bq_1; at $j = 2$, new directions in vectors A^2q_1 first and then BAq_1; at $j = 3$ new directions in vectors ABq_1 and then B^2q_1; and so on. The following table displays new vectors that expand the same subspace as the vectors generated at step j.

	j						
	1	2	3	4	5	6	7
New Vectors	Aq_1 Bq_1	A^2q_1 BAq_1	ABq_1 B^2q_1	A^3q_1 BA^2q_1	$ABAq_1$ B^2Aq_1	A^2Bq_1 $BABq_1$	AB^2q_1 B^3q_1

We list those vectors in the order of their first appearances as

$$\begin{array}{ll} \text{Group 0:} & q_1, \\ \text{Group 1:} & Aq_1, Bq_1, \\ \text{Group 2:} & A^2q_1, BAq_1, ABq_1, B^2q_1, \\ \text{Group 3:} & A^3q_1, BA^2q_1, ABAq_1, B^2Aq_1, A^2Bq_1, BABq_1, AB^2q_1, B^3q_1, \\ \vdots & \vdots \end{array} \tag{2.7}$$

They are produced in the order from top downwards and from left to right, and recursively if renamed as $f_0 = q_1, f_1, f_2, \ldots$, then $f_{2k-1} = Af_{k-1}$ and $f_{2k} = Bf_{k-1}$ for $k \geqslant 1$, Notice that we divide them naturally into Groups with Group 0 having only q_1, and Group t having 2^t vectors in the forms

$$X_t \cdots X_2 X_1 q_1, \quad X_i \in \{A, B\}.$$

The rule that governs the ordering in (2.7) is

$$\begin{aligned} &X_s \cdots X_2 X_1 q_1 \text{ appears before } Y_t \cdots Y_2 Y_1 q_1 \text{ in (2.7) if } s < t \text{ or} \\ &\quad \text{if there is an } j\ (1 \leqslant j \leqslant t) \text{ such that } X_i = Y_i \text{ for } 1 \leqslant i \leqslant j-1, \\ &\quad X_j = A, \text{ and } Y_j = B \text{ when } s = t. \end{aligned} \tag{2.8}$$

Define

$$\mathrm{g}\mathscr{K}_\ell(\{A, B\}, q_1) \tag{2.9}$$

to be the subspace spanned by the first ℓ vectors in (2.7) equipped with the ordering just described. We call it a *generalized Krylov subspace*.

Now in the notation of Algorithm 2.1, if $h_{a;N+1,j} = 0$ (or $h_{b;N+1,j} = 0$), then the first $N+1$ vectors in (2.7) are linearly dependent. In that case, q_{N+1} is constructed by using the next linearly independent vector, say the ℓth vector, in the sequence (2.7). At that point, the number of linearly independent vectors in the first ℓ vectors is exactly $N+1$. This leads to the following general result.

Theorem 2.1

(1) *If* $\dim \mathrm{g}\mathscr{K}_\ell(\{A, B\}, q_1) = N$, *then*

$$\mathrm{span}\{q_1, q_2, \ldots, q_N\} = \mathrm{g}\mathscr{K}_\ell(\{A, B\}, q_1).$$

(2) *If the* j*-loop of Algorithm* 2.1 *runs to its completion*, *then*

$$\mathrm{span}\{q_1, q_2, \ldots, q_N\} = \mathrm{g}\mathscr{K}_{2k+1}(\{A, B\}, q_1).$$

(3) *If Algorithm* 2.1 *concludes by* **BREAK***ing out at Line* 4, *then for* $\ell \geqslant 2j-1$,

$$\mathrm{span}\{q_1, q_2, \ldots, q_N\} = \mathrm{g}\mathscr{K}_\ell(\{A, B\}, q_1).$$

Proof. The first two claims are rather obvious. We shall now prove the third one. We have

$$Aq_j = \sum_{i=1}^{\alpha_j} q_i h_{a;ij} \quad \text{and} \quad Bq_j = \sum_{i=1}^{\beta_j} q_i h_{b;ij}.$$

When the **BREAK**ing out occurs, $\beta_j = j$ as β_j is the value of N. Together with $\alpha_j \leqslant \beta_j$, this shows that $\text{span}\{q_1, q_2, \ldots, q_N\}$ is an invariant subspace for both A and B and is the same as $\text{g}\mathscr{K}_{2j-1}(\{A, B\}, q_1)$. The vectors from the $2j$th onwards in (2.7) are linear combinations of vectors from the first to the $(2j-1)$th multiplied by sequences of A and/or B and thus fall into $\text{g}\mathscr{K}_{2j-1}(\{A, B\}, q_1)$. □

2.3. The case when A and B commute

When A and B commute, the reduced H_a and H_b will have much fewer nonzero entries below the diagonal than those for the general case in (2.1). We note that, for the eigenvalue problem $I\lambda^2 - A\lambda - B$, A and B share the same invariant subspace and thus we can just run the standard Arnoldi/Lanczos process on either A or B and then solve the quadratic eigenvalue problem. However, for other problems like the reduced order modelling to evaluate $q_1^*(I - As - Bs^2)^{-1}q_1$, the new processes may still be of interest. This subsection also shows that some inherent relations between A and B will have interesting effects on our new Arnoldi type process (and the Lanczos type process later in this paper).

The commutativity between A and B implies that some vectors in (2.7) appear multiple times, i.e., $BAq_1 = ABq_1$. In fact, Group t which has 2^t vectors effectively consists of $t+1$ vectors

$$A^t q_1, BA^{t-1}q_1, \ldots, B^{t-1}Aq_1, B^t q_1$$

in the generic situation. It can be seen that the generated basis vectors

$$q_1, q_2, q_3, \ldots,$$

correspond to the sequence

$$\begin{array}{ll} \text{Group 0:} & q_1, \\ \text{Group 1:} & Aq_1, Bq_1, \\ \text{Group 2:} & A^2q_1, BAq_1, B^2q_1, \\ \text{Group 3:} & A^3q_1, BA^2q_1, B^2Aq_1, B^3q_1, \\ \vdots & \vdots \end{array} \tag{2.10}$$

in the sense that the new direction in, e.g., q_5 is from BAq_1. We shall use "$\sim$" to indicate such a correspondence, e.g., $q_5 \sim BAq_1, q_8 \sim BA^2q_1$, and so on. We would like to know the nonzero patterns in the generated H_a and H_b. It suffices for us to look at Aq_j and Bq_j and find out the first positions at which the vectors in (2.10) bring out the same new directions as Aq_j and Bq_j do for expanding the generalized

Krylov subspace. First we need to know the corresponding position for q_j in (2.10). To this end, let integers s and t be such that

$$j = t(t+1)/2 + s \quad \text{for some } 0 < s \leqslant t+1,$$

i.e., q_j belongs to Group t in (2.10), and $q_j \sim B^{s-1}A^{t-(s-1)}q_1$. Thus

$$Aq_j \sim B^{s-1}A^{t+1-(s-1)}q_1 \sim q_{(t+1)(t+2)/2+s} = q_{j+t+1}, \tag{2.11}$$

$$Bq_j \sim B^{s}A^{t+1-s}q_1 \sim q_{(t+1)(t+2)/2+s+1} = q_{j+t+2}. \tag{2.12}$$

So the jth column of H_a has nonzero entries from position 1 to $j+t+1$ and the jth column of H_b has nonzero entries from position 1 to $j+t+2$. Fig. 1 shows the structures of H_a and H_b for the commutative case.

Theorem 2.2. *Given $q_1 \in \mathbb{C}^n$ with $\|q_1\|_2 = 1$, suppose that A and B commute and that the first n vectors in (2.10) are linearly independent. Then there is a unitary matrix $Q \in \mathbb{C}^{n\times n}$ with $Qe_1 = q_1$ such that*

$$Q^*AQ = H_a \equiv (h_{a;ij}), \quad Q^*BQ = H_b \equiv (h_{b;ij}) \tag{2.13}$$

satisfy

$$h_{a;ij} = 0 \text{ for } i \geqslant j+t+2, \quad h_{b;ij} = 0 \text{ for } i \geqslant j+t+3, \tag{2.14}$$

where t is the unique integer such that $t(t+1)/2 + (t+1) \geqslant j > t(t+1)/2$.

It is of interest to compare nonzero patterns for H_a and H_b here with those for H_a and H_b in (2.1) for general A and B, where there are about j nonzero entries below the diagonal entries in the jth columns. Theorem 2.2, however, says when A and B commute there are about $\sqrt{2j}$ nonzero entries below the diagonal entries since $t \approx \sqrt{2j}$ for large j.

It is of an independent interest to see how many trips to the j-loop in Algorithm 3.1 for commutative A and B must be made in order to produce an orthonormal basis for the entire space $\mathbb{C}^n$, assuming that the first n vectors in (2.10) are linearly independent. (Recall that in the generic case and noncommutative A and B, $(n-1)/2$ trips are enough.) To this end, by (2.11) and (2.12) we need to find the minimal j so that

$$j+t+1 = n \quad \text{or} \quad j+t+2 = n,$$
$$\text{subject to } j = t(t+1)/2 + s, \quad 1 \leqslant s \leqslant t+1,$$

or equivalently $t(t+1)/2 + 1 \leqslant j \leqslant t(t+1)/2 + t + 1$. Write $j+t+2 = \hat{n}$, where $\hat{n} = n$ or $n+1$ as needed. Then

$$t(t+1)/2 + t + 3 \leqslant \hat{n} \leqslant t(t+1)/2 + 2t + 3$$
$$\iff t^2 + 3t + 6 - 2\hat{n} \leqslant 0 \leqslant t^2 + 5t + 6 - 2\hat{n},$$

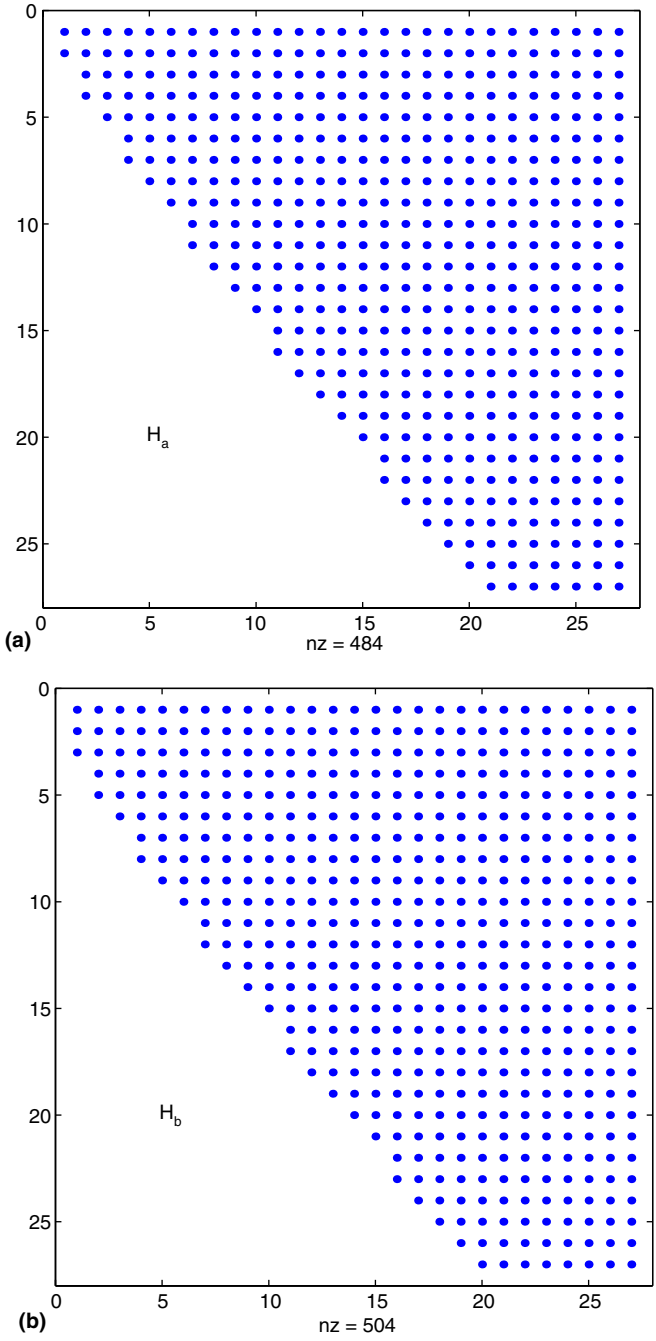


Fig. 1. The sparsity patterns of H_a and H_b: the commutative case.

which gives

$$\frac{-5+\sqrt{8\hat{n}+1}}{2} \leqslant t \leqslant \frac{-3+\sqrt{8\hat{n}-15}}{2}.$$

Thus

$$\hat{n} - \frac{1}{2} - \frac{\sqrt{8\hat{n}-15}}{2} \leqslant j \leqslant \hat{n} + \frac{1}{2} - \frac{\sqrt{8\hat{n}+1}}{2}.$$

For example when $n = 1000$, this requires $j = 955$. Asymptotically it needs n steps.

2.4. General monic matrix polynomial

The Arnoldi type process developed for A and B can be extended to a general monic matrix polynomial (1.1) of degree m. Recall the theoretical backbone of the algorithm is the decomposition (2.1). So we shall just state a corresponding decomposition result for $A(\lambda)$ but omit a detailed statement of an algorithm. The actual algorithm follows readily from this.

Theorem 2.3. *Let $A_\ell \in \mathbb{C}^{n\times n}$, $0 \leqslant \ell \leqslant m-1$. Given $q_1 \in \mathbb{C}^n$ with $\|q_1\|_2 = 1$, there is a unitary matrix $Q \in \mathbb{C}^{n\times n}$ with $Qe_1 = q_1$ such that*

$$Q^* A_\ell Q = H_\ell \equiv (h_{\ell;ij}), \quad \text{for } 0 \leqslant \ell \leqslant m-1 \tag{2.15}$$

satisfying

$$h_{\ell;ij} = 0, \quad \text{for } i \geqslant mj + m - \ell. \tag{2.16}$$

3. Lanczos type process for monic quadratic matrix polynomials

In this section, we develop a Lanczos type processes in parallel to the Arnoldi type process presented in the previous section. We shall present the derivation for monic quadratic matrix polynomial $I\lambda^2 - A\lambda - B$, and indicate a generalization to general mth degree monic matrix polynomials.

3.1. Lanczos type process

Given $v_1, w_1 \in \mathbb{C}^n$ with $w_1^* v_1 = 1$, a sequence of similarity transformations V_1, $V_2, \cdots, V_k$ ($k \leqslant n$) can be constructed (unless a division by zero is encountered in the process) such that

$$V^{-1} A V = T_a \equiv (t_{a;ij}), \quad V^{-1} B V = T_b \equiv (t_{b;ij}) \tag{3.1}$$

with

$$\begin{aligned} t_{a;ij} &= 0, \quad \text{for } i \geqslant 2j+1 \text{ or } j \geqslant 2i+1, \\ t_{b;ij} &= 0, \quad \text{for } i \geqslant 2j+2 \text{ or } j \geqslant 2i+2. \end{aligned} \tag{3.2}$$

where $V = V_1 V_2 \cdots V_k$ satisfies $Ve_1 = v_1$ and $V^{-*}e_1 = w_1$, see [16] for the construction. Eq. (3.1) or its partial reduction can also be realized by a Lanczos type process, which we present now.

Let $W = V^{-*}$ and rewrite (3.1) to get

$$\begin{aligned} &AV = VT_a, \quad A^*W = WT_a^*, \quad W^*V = I_n, \\ &BV = VT_b, \quad B^*W = WT_b^*. \end{aligned} \tag{3.3}$$

Notice that v_1 and w_1 are given and $w_1^* v_1 = 1$. We shall show how to progressively compute the first few columns of $V = (v_1, v_2, \ldots, v_n)$, $W = (w_1, w_2, \ldots, w_n)$, T_a and T_b from v_1 and w_1 until a breakdown occurs. We have

$$Av_1 = v_1 t_{a;11} + v_2 t_{a;21}, \tag{3.4}$$

$$A^* w_1 = w_1 \bar{t}_{a;11} + w_2 \bar{t}_{a;12}, \tag{3.5}$$

$$Bv_1 = v_1 t_{b;11} + v_2 t_{b;21} + v_3 t_{b;31}, \tag{3.6}$$

$$B^* w_1 = w_1 \bar{t}_{b;11} + w_2 \bar{t}_{b;12} + w_3 \bar{t}_{b;13}. \tag{3.7}$$

Eqs. (3.4) and (3.5), and biorthogonality between V's columns and W's columns yield $t_{a;11} = w_1^* A v_1$. Set

$$\hat{v}_2 = Av_1 - v_1 t_{a;11}, \quad \hat{w}_2 = A^* w_1 - w_1 \bar{t}_{a;11}.$$

We may take $t_{a;21} = \sqrt{|\hat{w}_2^* \hat{v}_2|}$. A breakdown occurs if $t_{a;21} = 0$; otherwise we assign

$$t_{a;12} = \hat{w}_2^* \hat{v}_2 / t_{a;21}, \quad v_2 = \hat{v}_2 / t_{a;21}, \quad w_2 = \hat{w}_2 / \bar{t}_{a;12}.$$

Then, $w_2^* v_2 = 1$. Next we turn to (3.6) and (3.7). Analogously we obtain

$$t_{b;11} = w_1^* B v_1, \quad t_{b;21} = w_2^* B v_1, \quad t_{b;12} = w_1^* B v_2. \tag{3.8}$$

Set

$$\hat{v}_3 = Bv_1 - v_1 t_{b;11} - v_2 t_{b;21}, \quad \hat{w}_3 = B^* w_1 - w_1 \bar{t}_{b;11} - w_2 \bar{t}_{b;12}. \tag{3.9}$$

We may take $t_{b;31} = \sqrt{|\hat{w}_3^* \hat{v}_3|}$. A breakdown occurs if $t_{b;31} = 0$; otherwise we assign

$$t_{b;13} = \hat{w}_3^* \hat{v}_3 / t_{b;31}, \quad v_3 = \hat{v}_3 / t_{b;31}, \quad w_3 = \hat{w}_3 / \bar{t}_{b;13}.$$

Then, $w_3^* v_3 = 1$. In general, we have for $j \geqslant 2$:

$$Av_j = \sum_{i=\lfloor (j+1)/2 \rfloor}^{2j-1} v_i t_{a;ij} + v_{2j} t_{a;2jj}, \tag{3.10}$$

$$A^* w_j = \sum_{i=\lfloor (j+1)/2 \rfloor}^{2j-1} w_i \bar{t}_{a;ji} + w_{2j} \bar{t}_{a;j2j}, \tag{3.11}$$

$$Bv_j = \sum_{i=\lfloor j/2 \rfloor}^{2j} v_i t_{b;ij} + v_{2j+1} t_{b;2j+1j}, \tag{3.12}$$

$$B^* w_j = \sum_{i=\lfloor j/2 \rfloor}^{2j} w_i \bar{t}_{b;ji} + w_{2j+1} \bar{t}_{b;j2j+1}. \tag{3.13}$$

with similar expressions for B and B^*. Eqs. (3.10) and (3.11), and biorthogonality between V's columns and W's columns yield

$$t_{a;ij} = w_i^* A v_j, \quad t_{a;ji} = w_j^* A v_i, \quad \text{for } \lfloor (j+1)/2 \rfloor \leqslant i \leqslant 2j-1. \tag{3.14}$$

Set

$$\hat{v}_{2j} = A v_j - \sum_{i=\lfloor (j+1)/2 \rfloor}^{2j-1} v_i t_{a;ij}, \quad \hat{w}_{2j} = A^* w_j - \sum_{i=\lfloor (j+1)/2 \rfloor}^{2j-1} w_i \bar{t}_{a;ji}. \tag{3.15}$$

We may take $t_{a;2jj} = \sqrt{|\hat{w}_{2j}^* \hat{v}_{2j}|}$. A breakdown occurs if $t_{a;2jj} = 0$; otherwise we assign

$$t_{a;j2j} = \hat{w}_{2j}^* \hat{v}_{2j} / t_{a;2jj}, \quad v_{2j} = \hat{v}_{2j} / t_{a;2jj}, \quad w_{2j} = \hat{w}_{2j} / \bar{t}_{a;j2j}.$$

Then, $w_{2j}^* v_{2j} = 1$. Similarly we get v_{2j+1} and w_{2j+1} from Bv_j and $B^* w_j$, respectively. Analogously we obtain

$$t_{b;ij} = w_i^* B v_j, \quad t_{b;ji} = w_j^* B v_i, \quad \text{for } \lfloor j/2 \rfloor \leqslant i \leqslant 2j. \tag{3.16}$$

Set

$$\hat{v}_{2j+1} = B v_j - \sum_{i=\lfloor j/2 \rfloor}^{2j} v_i t_{b;ij}, \quad \hat{w}_{2j+1} = B^* w_j - \sum_{i=\lfloor j/2 \rfloor}^{2j} w_i \bar{t}_{b;ji}. \tag{3.17}$$

We may take $t_{b;2j+1j} = \sqrt{|\hat{w}_{2j+1}^* \hat{v}_{2j+1}|}$. A breakdown occurs if $t_{b;2j+1j} = 0$; otherwise we assign

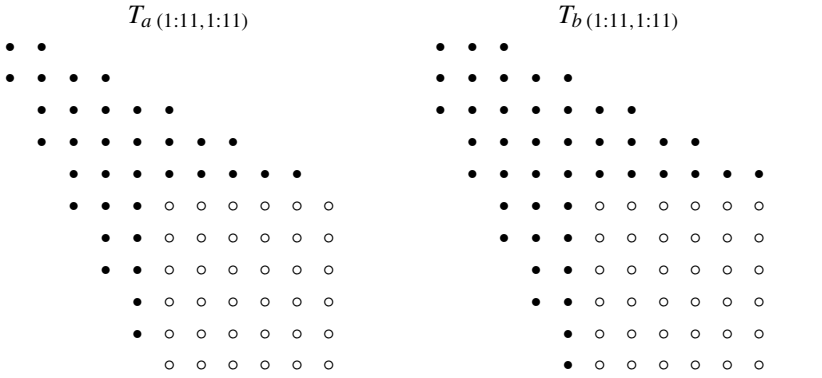
$$t_{b;j2j+1} = \hat{w}_{2j+2}^* \hat{v}_{2j+1} / t_{b;2j+1j}, \quad v_{2j+1} = \hat{v}_{2j+1} / t_{b;2j+1j},$$
$$w_{2j+1} = \hat{w}_{2j+1} / \bar{t}_{b;j2j+1}.$$

Then, $w_{2j+1}^* v_{2j+1} = 1$.

If A and B are Hermitian and $v_1 = w_1$, then T_a and T_b are symmetric and $v_i = w_i$, and the above computation is equivalent to the Arnoldi type process for the symmetric problem.

The following figures in (3.18) show what the computed part of T_a and T_b look like for j, as in (3.10)–(3.13), running from 1 to $k = 5$, where the entries marked by *unfilled circles* have not been computed yet, but they can be readily computed afterwards by

$$t_{a;ij} = w_i^* A v_j, \quad t_{b;ij} = w_i^* B v_j, \quad \text{for } k+1 \leqslant i, j \leqslant 2k+1.$$



(3.18)

We now consider a benign case of breakdown $t_{a;2jj} = 0$; that is when $\hat{v}_{2j} = \hat{w}_{2j} = 0$ (see (3.15)). Then, Av_j is linearly dependent on the v_i's that is already generated and A^*w_j is also linearly dependent on w_i's already generated. In this case, the process can be continued by using Av_{j+1} and A^*w_{j+1}, or the later vectors in the sequence, to construct v_{2j+1} and w_{2j+1}. The case that $t_{b;2j+1j} = 0$ with $\hat{v}_{2j+1} = \hat{w}_{2j+1} = 0$ in (3.17) is treated similarly. This leads to Algorithm 3.1. We also note that, when $\hat{v}_{2j} = 0$ but $\hat{w}_{2j} \neq 0$ (another case of breakdown), the process can also be continued by assigning to $\hat{v}_{2j}$ any vector that is orthogonal to all w_i generated but not to $\hat{w}_{2j}$, but we shall leave the detail along this line to future study.

Algorithm 3.1 (Lanczos Type Process)

1. **Given** v_1 **and** w_1 **such that** $w_1^* v_1 = 1$;
2. $N = 1$; $\alpha_1 = 1$; $\beta_1 = 1$; $\ell_a = 1$; $\ell_b = 1$;
3. **for** $j = 1, 2, \ldots, k$ **do**
4. **if** $j > N$ **then BREAK**;
5. $\hat{v} = Av_j$; $\hat{w} = A^*w_j$;
6. **if** $j > \alpha_{\ell_a}$ **then** $\ell_a = \ell_a + 1$;
7. **for** $i = \ell_a, \ldots, N$ **do**
8. $t_{a;ij} = w_i^* \hat{v}$; $\hat{v} = \hat{v} - v_i t_{a;ij}$;
9. $t_{a;ji} = \hat{w}^* v_i$; $\hat{w} = \hat{w} - w_i \bar{t}_{a;ji}$;
10. **end for**
11. $t_{a;N+1,j} = \sqrt{|\hat{w}^* \hat{v}|}$;
12. **if** $t_{a;N+1,j} = 0$ **then**
13. **if** $\hat{v} \neq 0$ **or** $\hat{w} \neq 0$ **then BREAK**;
14. **else**
15. $N = N + 1; t_{a;jN} = \hat{w}^* \hat{v} / t_{a;Nj}; v_N = \hat{v} / t_{a;Nj}; w_N = \hat{w} / \bar{t}_{a;jN}, \alpha_j = N$;
16. **end if**

17. $\hat{v} = Bv_j$; $\hat{w} = B^* w_j$;
18. **if** $j > \beta_{\ell_b}$ then $\ell_b = \ell_b + 1$;
19. **for** $i = \ell_b, \ldots, N$ **do**
20. $t_{b;ij} = w_i^* \hat{v}$; $\hat{v} = \hat{v} - v_i t_{b;ij}$;
21. $t_{b;ji} = \hat{w}^* v_i$; $\hat{w} = \hat{w} - w_i \bar{t}_{b;ji}$;
22. **end for**
23. $t_{b;N+1,j} = \sqrt{|\hat{w}^* \hat{v}|}$;
24. **if** $t_{b;N+1,j} = 0$ **then**
25. **if** $\hat{v} \neq 0$ **or** $\hat{w} \neq 0$ **then BREAK**;
26. **else**
27. $N = N + 1, t_{b;jN} = \hat{w}^* \hat{v} / t_{b;Nj}$; $v_N = \hat{v} / t_{b;Nj}$; $w_N = \hat{w} / \bar{t}_{b;jN}$, $\beta_j = N$;
28. **end if**
29. **end for**

In Algorithm 3.1, N tracks the number of vectors v_i already constructed at any given point, which is also the number of vectors w_i already constructed at that point; α_j is the value of N at the end of Line 15, i.e., the row number of the last nonzero entry in the jth column of T_a and β_j is the value of N at the end of Line 27, i.e., the row number of the last nonzero entry in the jth column of T_b; ℓ_a (and ℓ_b) tracks the row number of the first nonzero entry in the jth column of T_a (and T_b resp.). Then ℓ_a is the smallest integer such that $\alpha_{\ell_a} \geqslant j$, and ℓ_b is the smallest integer such that $\beta_{\ell_b} \geqslant j$.

If the execution of Algorithm 3.1 is completed by the **BREAK** statement at Line 4, a common (right) invariant subspace and a common left invariant subspace for A and B have been computed, and

$$AV_{(:,1:N)} = V_{(:,1:N)} T_{a(1:N,1:N)}, \quad A^* W_{(:,1:N)} = W_{(:,1:N)} [T_{a(1:N,1:N)}]^*, \tag{3.19}$$

$$BV_{(:,1:N)} = V_{(:,1:N)} T_{b(1:N,1:N)}, \quad B^* W_{(:,1:N)} = W_{(:,1:N)} [T_{b(1:N,1:N)}]^*, \tag{3.20}$$

If none of the **BREAK** statements is executed, we have

$$AV_{(:,1:k)} = V_{(:,1:\alpha_k)} T_{a(1:\alpha_k,1:k)}, \quad A^* W_{(:,1:k)} = W_{(:,1:\alpha_k)} [T_{a(1:k,1:\alpha_k)}]^*, \tag{3.21}$$

$$BV_{(:,1:k)} = V_{(:,1:\beta_k)} T_{b(1:\beta_k,1:k)}, \quad B^* W_{(:,1:k)} = W_{(:,1:\beta_k)} [T_{b(1:k,1:\beta_k)}]^*, \tag{3.22}$$

In this case, parts of projections T_a and T_b are not computed, but they can be readily obtained afterwards by

$$t_{a;ij} = q_i^* A q_j, \quad t_{b;ij} = q_i^* B q_j, \quad \text{for } k+1 \leqslant i \leqslant N \text{ and } k+1 \leqslant j \leqslant N.$$

It can be seen that the nonzero patterns, i.e., the positions of nonzero entries, in T_a and T_b here, are contained in those as described in (3.18).

The following theorem shows that indeed two sets $\{w_i\}$ and $\{v_i\}$ by Algorithm 3.1 enjoy the desired biorthogonality property.

Theorem 3.1. *Suppose Algorithm 3.1 runs to its completion without breakdowns to produce* $\{w_i\}_{i=1}^N$ *and* $\{v_i\}_{i=1}^N$, *where* $N = \beta_k$. *Then*

$$w_i^* v_j = 0 \quad \text{if } i \neq j \text{ and } w_i^* v_i = 1.$$

Proof. $w_i^* v_i = 1$ is clear from their definition. We shall only need to show $w_i^* v_j = 0$ if $i \neq j$. Let ℓ_{aj} and ℓ_{bj} be the ℓ_a and ℓ_b used at Lines 7 and 19, respectively, at the jth trip of the j-loop, $\hat{v}_i$ and $\hat{w}_i$ be the ones finally used to give v_i and w_i by scalar scaling in Algorithm 3.1.

We prove the claim by induction on k. The case for $k = 1$ is easy to establish. Suppose the claim in the theorem holds for $k = j - 1$. We now show it also holds for $k = j$, i.e., $w_s^* \hat{v}_{\alpha_j} = 0 = \hat{w}_{\alpha_j}^* v_s$ for $s < \alpha_j$ and $w_s^* \hat{v}_{\beta_j} = 0 = \hat{w}_{\beta_j}^* v_s$ for $s < \beta_j$. $w_s^* \hat{v}_{\alpha_j} = 0 = \hat{w}_{\alpha_j}^* v_s$ for $\ell_{aj} \leqslant s \leqslant \alpha_j - 1$ follows from the definition. For the case of $s < \ell_{aj}$, we have

$$\begin{aligned}
w_s^* \hat{v}_{\alpha_j} &= w_s^* \left(A v_j - \sum_{i=\ell_{aj}}^{\alpha_j - 1} v_i t_{a;ij} \right) \\
&= w_s^* A v_j = \left[\hat{w}_{\alpha_s} + \sum_{i=\ell_{as}}^{\alpha_s - 1} w_i \bar{t}_{a;si} \right]^* v_j \\
&= 0, \\
\hat{w}_{\alpha_j}^* v_s &= \left[A^* w_j - \sum_{i=\ell_{aj}}^{\alpha_j - 1} w_i \bar{t}_{a;ji} \right]^* v_s \\
&= w_j^* A v_s = w_j^* \left(\hat{v}_{\alpha_s} + \sum_{i=\ell_{as}}^{\alpha_s - 1} v_i t_{a;is} \right) \\
&= 0,
\end{aligned}$$

by induction hypothesis, since $s < \ell_{aj}$ implies $\alpha_s < j$ by Line 6. Similarly we have $\hat{w}_{\alpha_j}^* v_s = 0$.

Analogously we can show $w_s^* \hat{v}_{\beta_j} = 0 = \hat{w}_{\beta_j}^* v_s$ for $s < \beta_j$. □

3.2. General monic matrix polynomial

As in Section 2.4, the Lanczos type process developed can also be extended to a general monic matrix polynomial (1.1) of degree m. Again, we just state the corresponding decomposition results and omit a detailed statement of an algorithm.

Theorem 3.2. *Let $A_\ell \in \mathbb{C}^{n\times n}$, $0 \leqslant \ell \leqslant m-1$. Given $v_1, w_1 \in \mathbb{C}^n$ such that $w_1^* v_1 = 1$, (unless there is a breakdown) there is a matrix $V \in \mathbb{C}^{n\times n}$ with $Ve_1 = v_1$ and $V^{-*}e_1 = w_1$ such that*

$$V^{-1}A_\ell V = T_a \equiv (t_{\ell;ij}), \quad \text{for } 0 \leqslant \ell \leqslant m-1 \tag{3.23}$$

satisfying

$$t_{\ell;ij} = 0, \quad \text{for } i \geqslant mj + m - \ell \ \text{ or } \ j \geqslant mi + m - \ell. \tag{3.24}$$

4. Application to model reduction

In a second order single-input and single-output linear system, a quadratic matrix polynomial is involved in its transfer function

$$f(s) = c^*(I - As - Bs^2)^{-1}b$$

where b and c are n-dimensional vectors, A and B are $n \times n$, either sparse or in some kinds of factored forms. In model reductions, it is desirable that the given system is approximated by another second order system of lower dimension that is called a reduced system. The approximation is usually in terms of the transfer functions and is often done by requiring that the transfer function of the reduced system $g(s)$ and the original transfer function $f(s)$ to have the same moments up to certain degree (i.e., terms associated with $s^0, s^1, s^2, \ldots$ of their Taylor expansions at $s = 0$). In the case of first order systems, Feldman and Freund [10] show that the Lanczos algorithm is a powerful method that can be used to achieve this. Here we shall show that the Arnoldi/Lanczos type algorithms derived in the previous sections can be used in the same way for second order systems.

- For the Arnoldi type process with $q_1 = b/\|b\|_2$, let Algorithm 2.1 produce $Q_{(:,1:N)}$, $H_{a(1:N,1:N)}$, and $H_{b(1:N,1:N)}$. Define

$$g_{\text{arnd}}(s) = \tilde{c}_N^* \left(I - H_{a(1:N,1:N)}s - H_{b(1:N,1:N)}s^2\right)^{-1} \|b\|_2 e_1, \tag{4.1}$$

  where $\tilde{c}_N = Q_{(:,1:N)}^* c$.
- For the Lanczos type process with $v_1 = b/\|b\|_2$ and $w_1 = c/b^*c$, suppose that Algorithm 3.1 runs to its completion without breakdowns and produces $V_{(:,1:N)}$, $W_{(:,1:N)}$, $T_{a(1:N,1:N)}$, and $T_{b(1:N,1:N)}$. Define

$$g_{\text{lanz}}(s) = (c^*b)e_1^* \left(I - T_{a(1:N,1:N)}s - T_{b(1:N,1:N)}s^2\right)^{-1} e_1. \tag{4.2}$$

We shall next find out how accurate $g_{\text{arnd}}(s)$ and $g_{\text{lanz}}(s)$ are as approximations to $f(s)$ by determining the numbers of matching leading terms in their Taylor expansions at $s = 0$. We start by presenting a lemma.

Lemma 4.1. *Let $X = (x_{ij})$ and $Y = (y_{ij})$ be two $n \times n$ matrices satisfying*

$$x_{ij} = 0, \quad \text{for } i > k_1 j + \ell_1 \quad \text{and} \quad y_{ij} = 0, \quad \text{for } i > k_2 j + \ell_2,$$

and let $Z = XY = (z_{ij})$. Assume that no zero entries in Z are caused by exact arithmetic cancellations. Then $z_{ij} = 0$ if and only if $i > k_1 k_2 j + (k_1 \ell_2 + \ell_1)$.

Proof. We have

$$\begin{aligned} z_{ij} = \sum_{m=1}^{n} x_{im} y_{mj} &= \sum_{\{m: i \leqslant k_1 m + \ell_1\}} x_{im} y_{mj} + \sum_{\{m: i > k_1 m + \ell_1\}} x_{im} y_{mj} \\ &= \sum_{\{m: i \leqslant k_1 m + \ell_1\}} x_{im} y_{mj} + 0 \end{aligned} \tag{4.3}$$

since $x_{im} = 0$ for $i > k_1 m + \ell_1$. Now for $i > (k_1 k_2) j + (k_1 \ell_2 + \ell_1)$ and $i \leqslant k_1 m + \ell_1$, we have

$$k_1 m + \ell_1 > (k_1 k_2) j + (k_1 \ell_2 + \ell_1) \;\Rightarrow\; m > k_2 j + \ell_2;$$

so $y_{mj} = 0$. Therefore $z_{ij} = 0$ for $i > (k_1 k_2) j + (k_1 \ell_2 + \ell_1)$ by (4.3). It can be seen that in the generic case if $i \leqslant (k_1 k_2) j + (k_1 \ell_2 + \ell_1)$, then at least one summand in $\sum_{\{m: i \leqslant k_1 m + \ell_1\}} x_{im} y_{mj}$ is not zero. □

4.1. Arnoldi type process

Theorem 4.1. *Let $g_{\text{arnd}}(s)$ be as defined in (4.1) as a result of Algorithm 2.1. Then*

$$f(s) = g_{\text{arnd}}(s) + \mathcal{O}(s^{\lfloor \log_2 N \rfloor + 1}).$$

If, in addition, $c = b$, then $f(s) = g_{\text{arnd}}(s) + \mathcal{O}(s^{\lfloor \log_2 N \rfloor + 2})$.

This result is a bit of disappointing in terms of the order of approximation, compared unfavorably to the standard Arnoldi method for the linear model reduction which can achieve order $\mathcal{O}(s^k)$ of approximation [29]. But Theorem 4.1 appears to be best possible in general unless there are additional structures in A and B such as commutativity or the low rank case as in [18]. If the Arnoldi Type Process is performed to its completion, we will have $n \times n$ matrices Q, $H_a \equiv (h_{a;ij})$, and $H_b \equiv (h_{b;ij})$, satisfying (2.1) and (2.2). We note that the sparsity pattern of (2.2) defines an envelope

enclosing nonzeros of H_a and H_b, while the matrices H_a and H_b as generated by the algorithm may be more condensed. From the reduction, we have

$$f(s)/\|b\|_2 = \tilde{c}^*\big(I - H_a s - H_b s^2\big)^{-1} e_1,$$

where $\tilde{c} = Q^* c$. The Taylor expansion of $f(s)/\|b\|_2$ at $s = 0$ is

$$f(s)/\|b\|_2 = \tilde{c}^* \left(\sum_{\ell=0}^{\infty} H_\ell s^\ell\right) e_1 = \sum_{\ell=0}^{\infty} (\tilde{c}^* H_\ell e_1) s^\ell,$$

where, as a result of

$$\big(I - H_a s - H_b s^2\big)\left(\sum_{\ell=0}^{\infty} H_\ell s^\ell\right) = I_n = \left(\sum_{\ell=0}^{\infty} H_\ell s^\ell\right)\big(I - H_a s - H_b s^2\big),$$

H_ℓ can be recursively defined as

$$H_0 = I_n, \tag{4.4}$$
$$H_1 = H_a, \tag{4.5}$$
$$H_\ell = H_a H_{\ell-1} + H_b H_{\ell-2}, \quad \text{for } \ell \geqslant 2, \tag{4.6}$$
$$= H_{\ell-1} H_a + H_{\ell-2} H_b, \quad \text{for } \ell \geqslant 2. \tag{4.7}$$

On the other hand, it can be derived similarly that

$$g_{\text{arnd}}(s)/\|b\|_2 = \sum_{\ell=0}^{\infty} (\tilde{c}_N^* \widetilde{H}_\ell e_1) s^\ell,$$

where $\widetilde{H}_\ell$ is recursively defined as

$$\widetilde{H}_0 = I_N, \tag{4.8}$$
$$\widetilde{H}_1 = H_{a(1:N,1:N)}, \tag{4.9}$$
$$\widetilde{H}_\ell = H_{a(1:N,1:N)} \widetilde{H}_{\ell-1} + H_{b(1:N,1:N)} \widetilde{H}_{\ell-2}, \quad \text{for } \ell \geqslant 2, \tag{4.10}$$
$$= \widetilde{H}_{\ell-1} H_{a(1:N,1:N)} + \widetilde{H}_{\ell-2} H_{b(1:N,1:N)}, \quad \text{for } \ell \geqslant 2. \tag{4.11}$$

Lemma 4.2. *H_ℓ and $\widetilde{H}_\ell$ have the property that their (i, j)*th *entries are zero if $i > 2^\ell j$.*

Proof. We prove this by induction for H_ℓ; the proof is similar for $\widetilde{H}_\ell$. Clearly the result holds for $H_0 = I_n$ and $H_1 = H_a$. Suppose the result holds for all matrices $H_0, H_1, \ldots, H_\ell$, then the (i, j)th entry of $H_a H_\ell$ is zero for $i > 2^{\ell+1} j$ by Lemma 4.1 and by (2.2). It can be checked that the (i, j)th entry of $H_b H_{\ell-1}$ is zero for $i > 2^\ell j + 1$. But for $\ell \geqslant 1$ and $j \geqslant 1$, $2^{\ell+1} j \geqslant 2^\ell j + 2$. Therefore, the result now follows from the recurrence relation. □

Lemma 4.3. *Suppose $N \geqslant 3$, and let $m = \lfloor \log_2 N \rfloor$, the largest possible integer such that $2^m \leqslant N$. Then*

(1) $H_\ell e_j = \begin{matrix} N \\ n-N \end{pmatrix} \begin{pmatrix} \widetilde{H}_\ell e_j \\ 0 \end{pmatrix}$ *for $\ell = 0, 1, \ldots, m$ and $j = 1, 2, \ldots, 2^{m-\ell}$.*

(2) $H_{m+1} e_1 = \begin{matrix} N \\ n-N \end{pmatrix} \begin{pmatrix} \widetilde{H}_{m+1} e_1 \\ \mathrm{x} \end{pmatrix}$.

Here the integers to the left of the partitioned matrices are the row sizes of the corresponding blocks.

Proof. That the last $n - N$ entries of $H_\ell e_j$ for $0 \leqslant \ell \leqslant m$ and $1 \leqslant j \leqslant 2^{m-\ell}$ are all zeros is due to Lemma 4.2 since $N + 1 > 2^m = 2^\ell 2^{m-\ell} \geqslant 2^\ell j$. We shall prove Claim 1 by induction on ℓ. It holds true for $\ell = 0, 1$. Suppose $m \geqslant \ell \geqslant 2$ and that the claim holds for $0, 1, \ldots, \ell - 1$. Then

$$
\begin{aligned}
H_\ell e_j &= H_a H_{\ell-1} e_j + H_b H_{\ell-2} e_j \\
&= H_a \begin{pmatrix} \widetilde{H}_{\ell-1} e_j \\ 0 \end{pmatrix} + H_b \begin{pmatrix} \widetilde{H}_{\ell-2} e_j \\ 0 \end{pmatrix} \\
&= H_a \begin{pmatrix} \widetilde{H}_{\ell-1} & 0 \\ 0 & 0 \end{pmatrix} e_j + H_b \begin{pmatrix} \widetilde{H}_{\ell-2} & 0 \\ 0 & 0 \end{pmatrix} e_j \\
&= \begin{pmatrix} H_{a(1:N,1:N)} \widetilde{H}_{\ell-1} e_j \\ \mathrm{x} \end{pmatrix} + \begin{pmatrix} H_{b(1:N,1:N)} \widetilde{H}_{\ell-2} e_j \\ \mathrm{x} \end{pmatrix} \\
&= \begin{pmatrix} \widetilde{H}_\ell e_j \\ \mathrm{x} \end{pmatrix}.
\end{aligned}
$$

The lower block marked by x is 0 for $\ell \leqslant m$ and $j \leqslant 2^{m-\ell}$ as we commented; Claim 1 is proved. With Claim 1 proved, setting $\ell = m + 1$ and $j = 1$ in the above equations leads to Claim 2. □

Proof of Theorem 4.1. $\ell \leqslant m = \lfloor \log_2 N \rfloor$ implies $2^\ell \leqslant N$; and thus by Lemma 4.3

$$
\tilde{c}^* H_\ell e_1 = \begin{pmatrix} \tilde{c}_N \\ \mathrm{x} \end{pmatrix}^* \begin{pmatrix} \widetilde{H}_\ell e_1 \\ 0 \end{pmatrix} = \tilde{c}_N^* \widetilde{H}_\ell e_1,
$$

as expected. Now if $c = b$, $\tilde{c} = \tilde{c}_N = \|b\|_2 e_1$, then also by Lemma 4.3

$$
e_1^* H_{m+1} e_1 = \begin{pmatrix} e_1 \\ 0 \end{pmatrix}^* \begin{pmatrix} \widetilde{H}_{m+1} e_1 \\ \mathrm{x} \end{pmatrix} = e_1^* \widetilde{H}_{m+1} e_1,
$$

as was to be shown. □

4.2. Lanczos type process

Theorem 4.2. *Suppose Algorithm 3.1 runs to its completion without breakdowns with* $Ve_1 = \alpha b$ *and* $We_1 = \beta c$, *and as a result let* $g_{\text{lanz}}(s)$ *be as defined in (4.2). Assume also the process can be continued until all n columns of* V *(and* W*) are obtained. Then*

$$f(s) = g_{\text{lanz}}(s) + \mathcal{O}(s^{2\lfloor \log_2 N \rfloor + 1}).$$

Theorem 4.2 is a consequence of Claim 5 of Lemma 4.5.

The assumption that the Lanczos type process can be continued until all n columns of V (and W) are obtained is for our theoretical analysis only. In practice, the process will stop much earlier than that for large sparse matrices. With the assumption, we have V, W, $T_a \equiv (t_{a;ij})$, and $T_b \equiv (t_{b;ij})$, satisfying (3.1) and (3.2). Then,

$$f(s)/(c^*b) = e_1^*\big(I - T_a s - T_b s^2\big)^{-1} e_1$$

whose Taylor expansion at $s = 0$ is

$$f(s)/(c^*b) = e_1^* \left(\sum_{\ell=0}^{\infty} T_\ell s^\ell \right) e_1 = \sum_{\ell=0}^{\infty} (e_1^* T_\ell e_1) s^\ell,$$

where T_ℓ can be recursively defined the same as (4.4)–(4.7) with H replaced by T. On the other hand, for $g_{\text{lanz}}(s)$ we have

$$g_{\text{lanz}}(s)/(c^*b) = \sum_{\ell=0}^{\infty} (e_1^* \widetilde{T}_\ell e_1) s^\ell,$$

where $\widetilde{T}_\ell$ can also be recursively defined the same as (4.8)–(4.11) with H replaced by T.

Lemma 4.4. T_ℓ *and* $\widetilde{T}_\ell$ *have the property that their* (i, j)th *entries are zero if* $i > 2^\ell j$ *or* $2^\ell i < j$.

Proof. Analogous to the proof of Lemma 4.2. □

Lemma 4.5. *Suppose* $N \geqslant 3$, *and let* $m = \lfloor \log_2 N \rfloor$, *the largest possible integer such that* $2^m \leqslant N$. *Then*

(1) $T_\ell e_j = \begin{matrix} N \\ n - N \end{matrix} \begin{pmatrix} \widetilde{T}_\ell e_j \\ 0 \end{pmatrix}$ *for* $\ell = 0, 1, \ldots, m$ *and* $j = 1, 2, \ldots, 2^{m-\ell}$.

(2) $T_{m+1} e_1 = \begin{matrix} N \\ n - N \end{matrix} \begin{pmatrix} \widetilde{T}_{m+1} e_1 \\ \mathrm{x} \end{pmatrix}$.

(3) $e_i^* T_\ell e_1 = e_i^* \widetilde{T}_\ell e_1$ *for* $\ell = 0, 1, \ldots, 2m$ *and* $i = 1, 2, \ldots, \min\{2^m, 2^{2m-\ell}\}$.

Proof. Claims 1 and 2 follow from Lemma 4.3 since the nonzero patterns of T_a and T_b are contained in those of H_a and H_b. On the other hand, since the nonzero patterns of T_a^{T} and T_b^{T} are also contained in those of H_a and H_b, the conclusion of Lemma 4.3 applies to T_ℓ^{T}, which implies $e_i^* T_\ell = (e_i^* \widetilde{T}_\ell, 0)$ for $\ell = 0, 1, \ldots, m$ and $i = 1, 2, \ldots, 2^{m-\ell}$. and $e_1^* T_{m+1} = (e_1^* \widetilde{T}_{m+1}, 0)$.

We now prove Claim 3 by induction on ℓ. First for $0 \leqslant \ell \leqslant m$, the claim holds due to Claim 1; the claim also holds for $\ell = m + 1$ because of Claim 2. In fact, we can say more

$$e_i^* T_\ell e_1 = e_i^* \widetilde{T}_\ell e_1, \quad \text{for } 0 \leqslant \ell \leqslant m+1 \text{ and } 1 \leqslant i \leqslant 2^m.$$

We now prove Claim 5 for the rest of ℓ:

$$e_i^* T_\ell e_1 = e_i^* \widetilde{T}_\ell e_1, \quad \text{for } m \leqslant \ell \leqslant 2m \text{ and } 1 \leqslant i \leqslant 2^{2m-\ell}, \tag{4.12}$$

by induction on ℓ. This claim holds for $\ell = m, m+1$. Suppose that $m+2 \leqslant \ell \leqslant 2m$, $i \leqslant 2^{2m-\ell}$, and that (4.12) holds for $m, m+1, \ldots, \ell-1$. Notice that $2i \leqslant 2^{2m-(\ell-1)}$ and $2i + 1 < 2^{2m-(\ell-2)}$. Then

$$\begin{aligned} e_i^* T_\ell e_1 &= e_i^* (T_a T_{\ell-1} + T_b T_{\ell-2}) e_1 \\ &= \left(\sum_{j=\lfloor (i+1)/2 \rfloor}^{2i} t_{a;ij} e_j^* \right) T_{\ell-1} e_1 + \left(\sum_{j=\max\{1, \lfloor i/2 \rfloor\}}^{2i+1} t_{b;ij} e_j^* \right) T_{\ell-2} e_1 \\ &= \sum_{j=\lfloor (i+1)/2 \rfloor}^{2i} t_{a;ij} (e_j^* \widetilde{T}_{\ell-1} e_1) + \sum_{j=\max\{1, \lfloor i/2 \rfloor\}}^{2i+1} t_{b;ij} (e_j^* \widetilde{T}_{\ell-2} e_1), \end{aligned}$$

on using the induction hypothesis. This can be simplified to

$$\begin{aligned} e_i^* T_\ell e_1 &= e_i^* T_{a(1:N,1:N)} \widetilde{T}_{\ell-1} e_1 + e_i^* T_{b(1:N,1:N)} \widetilde{T}_{\ell-2} e_1 \\ &= e_i^* \widetilde{T}_\ell e_1, \end{aligned}$$

as expected. □

5. Quadratic eigenvalue problems

The Krylov type methods that we have derived can also be used to compute eigenvalues and eigenvectors of $I\lambda^2 - A\lambda - B$ as follows:

- For the Arnoldi type process, if Algorithm 2.1 produces $Q_{(:,1:N)}$, $H_{a(1:N,1:N)}$, and $H_{b(1:N,1:N)}$ and θ_i is an eigenvalue and u_i is a right eigenvector of

$$I\lambda^2 - H_{a(1:N,1:N)}\lambda - H_{b(1:N,1:N)}, \tag{5.1}$$

then we use (θ_i, x_i) as an approximate eigenpair for the original problem, where

$$x_i = Q_{(:,1:N)}u_i. \tag{5.2}$$

- For the Lanczos type process, if Algorithm 3.1 runs to its completion without breakdowns and produces $V_{(:,1:N)}$, $W_{(:,1:N)}$, $T_{a(1:N,1:N)}$, and $T_{b(1:N,1:N)}$, and if θ_i is an eigenvalue and u_i (and v_i) is a right (left, resp.) eigenvector of

$$I\lambda^2 - T_{a(1:N,1:N)}\lambda - T_{b(1:N,1:N)}, \tag{5.3}$$

then we use θ_i as an approximate eigenvalue with x_i (and y_i) as an approximate right (left, resp.) eigenvector for the original problem, where

$$x_i = V_{(:,1:N)}u_i, \quad y_i = W_{(:,1:N)}v_i. \tag{5.4}$$

The above processes approximate a quadratic eigenvalue problem with a projected quadratic eigenvalue problem.

We now discuss convergence properties for the Lanczos type process only. Corresponding results for the Arnoldi type process can be obtained similarly. We first present a residual bound for the Ritz values and vectors.

Theorem 5.1. *For the Ritz values and Ritz vector obtained by the Lanczos type process (Algorithm 3.1), we have*

$$\begin{aligned}&\|(I\theta_i^2 - A\theta_i - B)x_i\| \\ &\quad\leqslant \|V\| \left(\|\theta_i T_{a(N+1:\alpha_N,p:N)}\| + \|T_{b(N+1:\beta_N,p:N)}\|\right) \|u_{i,(p:N)}\|\end{aligned}$$

and

$$\begin{aligned}&\|y_i^*(I\theta_i^2 - A\theta_i - B)\| \\ &\quad\leqslant \|W\| \left(\|\theta_i T_{a(p:N,N+1:\alpha_N)}\| + \|T_{b(p:N,N+1:\beta_N)}\|\right) \|v_{i,(p:N)}\|,\end{aligned}$$

where p is the smallest integer such that $\beta_p > N$ and is equal to the value of ℓ_b at step $N+1$.

Proof. First, from (3.21) and (3.22), we have

$$\begin{aligned}AV_{(:,1:N)} &= V_{(:,1:N)}T_{a(1:N,1:N)} + V_{(:,N+1:\alpha_N)}T_{a(N+1:\alpha_N,1:N)}, \\ BV_{(:,1:N)} &= V_{(:,1:N)}T_{b(1:N,1:N)} + V_{(:,N+1:\beta_N)}T_{b(N+1:\beta_N,1:N)}\end{aligned}$$

Then

$$\left(I\theta_i^2 - A\theta_i - B\right)x_i = \left(\theta_i^2 V_{(:,1:N)} - \theta_i AV_{(:,1:N)} - BV_{(:,1:N)}\right)u_i$$

$$
\begin{aligned}
&= V_{(:,1:N)}\big(I\theta_i^2 - T_{a(1:N,1:N)}\theta_i - T_{b(1:N,1:N)}\big)u_i \\
&\quad - \theta_i V_{(:,N+1:\alpha_N)}T_{a(N+1:\alpha_N,1:N)}u_i - V_{(:,N+1:\beta_N)}T_{b(N+1:\beta_N,1:N)}u_i \\
&= -\theta_i V_{(:,N+1:\alpha_N)}T_{a(N+1:\alpha_N,p:N)}u_{i,(p:N)} \\
&\quad - V_{(:,N+1:\beta_N)}T_{b(N+1:\beta_N,p:N)}u_{i,(p:N)},
\end{aligned}
$$

where we note that the first $p-1$ columns of $T_{b(N+1:\beta_N,1:N)}$ and $T_{a(N+1:\alpha_N,1:N)}$ are zeros. Taking the norm of above, we obtain the bound. □

As in the standard Lanczos algorithm, bottom end elements of eigenvectors $u_{i,(p:N)}$ and $v_{i,(p:N)}$ typically become small for Ritz values in the extreme part of the spectrum and that leads to small residuals. In particular, the above bounds can be used to estimate the residual norms without actually computing them.

We next derive an error bound similar to the one for the nonsymmetric Lanczos algorithm derived in [30]. Let

$$
L = \begin{pmatrix} 0 & I \\ T_b & T_a \end{pmatrix} \quad \text{and} \quad L_N = \begin{pmatrix} 0 & I \\ T_{b(1:N,1:N)} & T_{a(1:N,1:N)} \end{pmatrix}.
$$

Lemma 5.1. *Let S_ℓ and T_ℓ be recursively defined by*

$$
\begin{aligned}
&S_0 = 0, \quad T_0 = I_n, \\
&S_1 = T_b, \quad T_1 = T_a, \\
&S_\ell = T_a S_{\ell-1} + T_b S_{\ell-2} T_\ell = T_a T_{\ell-1} + T_b T_{\ell-2}, \quad \text{for } \ell \geqslant 2.
\end{aligned}
$$

Then

$$
L^\ell = \begin{pmatrix} S_{\ell-1} & T_{\ell-1} \\ S_\ell & T_\ell \end{pmatrix}.
$$

Similarly, let $\widetilde{S}_\ell$ and $\widetilde{T}_\ell$ be recursively defined from L_N as in S_ℓ and T_ℓ above. Then,

$$
L_N^\ell = \begin{pmatrix} \widetilde{S}_{\ell-1} & \widetilde{T}_{\ell-1} \\ \widetilde{S}_\ell & \widetilde{T}_\ell \end{pmatrix}.
$$

Proof. It can be verified by induction. □

It can be seen that the conclusion of Lemma 4.5 holds for $\widetilde{T}_\ell$ and T_ℓ as well as for $\widetilde{S}_\ell$ and S_ℓ. Thus we have

$$
e_1^* S_\ell e_1 = e_1^* \widetilde{S}_\ell e_1,
$$

for $\ell = 0, 1, \ldots, 2m$ ($m = \lfloor \log_2 N \rfloor$), which implies

$$
e_1^* L^{\ell+1} e_1 = e_1^* L_N^{\ell+1} e_1.
$$

Therefore, for any polynomial f of degree $2m+1$,

$$
e_1^* f(L) e_1 = e_1^* f(L_N) e_1. \tag{5.5}
$$

There are other similar results. They include, for example,

$$e_{n+1}^* L^\ell e_1 = e_{k+1}^* L_N^\ell e_1. \tag{5.6}$$

For the sake of simplicity, we assume now that L and L_N are diagonalizable, and write

$$L_N = U^* \Theta V \quad \text{and} \quad L = X^* \Lambda Y, \tag{5.7}$$

where $\Theta = \operatorname{diag}(\theta_1, \ldots, \theta_m)$, $U^*V = I$ and $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$, $X^*Y = I$. Write $U = (u_{ij})$, $V = (v_{ij})$, $X = (x_{ij})$ and $Y = (y_{ij})$. Now substituting (5.7) into (5.5), we obtain

$$e_1^* X^* f(\Lambda) Y e_1 = e_1^* U^* f(\Theta) V e_1.$$

Thus

$$\sum_{i=1}^{n} f(\lambda_i) \bar{x}_{i1} y_{i1} = \sum_{i=1}^{m} f(\theta_i) \bar{u}_{i1} v_{i1}.$$

In particular, using $f(x) = (x - \theta_1) p(x)$, we have

$$\lambda_1 - \theta_1 = \frac{1}{p(\lambda_1)\bar{x}_{11} y_{11}} \left[- \sum_{i=2}^{n} (\lambda_i - \theta_1) p(\lambda_i) \bar{x}_{i1} y_{i1} + \sum_{i=2}^{m} (\theta_i - \theta_1) p(\theta_i) \bar{u}_{i1} v_{i1} \right].$$

This leads to the following theorem.

Theorem 5.2. *Let* $|\lambda_1 - \theta_1| = \min_j |\lambda_1 - \theta_j|$. *Then we have*

$$|\lambda_1 - \theta_1| \leqslant K \epsilon_{2m} \frac{\left(\sum_{i \neq 1} |x_{i1}|^2 + \sum_{i \neq 1} |u_{i1}|^2 \right)^{1/2}}{|x_{11}|} \frac{\left(\sum_{i \neq 1} |y_{i1}|^2 + \sum_{i \neq 1} |v_{i1}|^2 \right)^{1/2}}{|y_{11}|},$$

where

$$\epsilon_\ell = \min_{\deg(p) = \ell,\, p(\lambda_1) = 1} \max_{i \neq 1} \{ |p(\lambda_i)|, |p(\theta_i)| \}$$

and $K = \max_{i \neq 1} \{ |\lambda_i - \theta_1|, |\theta_i - \theta_1| \}$.

We note that, if λ_1 is an extreme eigenvalue, an appropriate polynomial can be chosen to make ϵ_ℓ small. Therefore, smaller bounds are obtained for the extreme part of the spectrum and for larger m. The last two terms in the bound depend on the angles between the initial vectors v_1 and w_1 and the right and left eigenvectors, respectively.

6. Numerical example

We present a numerical example in this section. Our example comes from a finite element discretization of the following problem from dissipative acoustics [6,28].

Let $\Omega \subset \mathbb{R}^2$ be a rectangular cavity filled with an acoustic fluid (such as air), with one absorbing wall Γ_A and three reflecting walls Γ_R. Let $P(x,t)$ and $U(x,t)$ be the acoustic pressure and the fluid displacement, respectively. Also let ρ be the density of the fluid, and c the speed at which the fluid conducts sound. Then the behavior of the fluid satisfies the equations

$$\rho \frac{\partial^2 U}{\partial t^2} + \nabla P = 0, \tag{6.1}$$

$$-\rho c^2 \operatorname{div} U = P, \tag{6.2}$$

with boundary conditions

$$U \cdot \nu = 0 \quad \text{on } \Gamma_R, \tag{6.3}$$

$$\alpha U \cdot \nu + \beta \frac{\partial U}{\partial t} \cdot \nu = P \quad \text{on } \Gamma_A, \tag{6.4}$$

where the scalar constants α, β are related to the impedance of the absorbing material. As in [6], we choose $\rho = 1$ kg/m^3, $c = 340$ m/s, $\alpha = 5 \times 10^4 4$ N/m^3, $\beta = 200$ N s/m^3 for our test; this choice of α and β models a very viscous absorbing material.

We are interested in finding the damped vibration modes of the fluid, which are solutions of the form $U(x,t) = e^{\lambda t} u(x)$, $P(x,t) = e^{\lambda t} p(x)$. Then, Eqs. (6.1)–(6.4) reduce to finding λ, p, u satisfying

$$\begin{aligned}
&\rho\lambda^2 u + \nabla p = 0 \quad \text{in } \Omega,\\
&p = -\rho c^2 \operatorname{div} u \quad \text{in } \Omega,\\
&p = (\alpha + \lambda\beta) u \cdot \nu \quad \text{on } \Gamma_A,\\
&u \cdot \nu = 0 \quad \text{on } \Gamma_R.
\end{aligned}$$

This system can be converted to a variational formulation. Let $\mathcal{V} = \{v \in H(\operatorname{div}, \Omega) : v \cdot \nu \in L^2(\partial\Omega) \text{ and } v \cdot \nu = 0 \text{ on } \Gamma_R\}$. The problem is equivalent to finding $\lambda \in \mathbb{C}$, nonzero $u \in \mathcal{V}$ so that

$$\lambda^2 \int_\Omega \rho u \cdot v + \lambda \int_{\Gamma_A} \beta u \cdot \nu v \cdot \nu + \int_{\Gamma_A} \alpha u \cdot \nu v \cdot \nu + \int_\Omega \rho c^2 \operatorname{div} u \operatorname{div} v = 0 \tag{6.5}$$

for all $v \in \mathcal{V}$. Using finite elements to approximate $\mathcal{V}$ by $\mathcal{V}_h = \operatorname{span}\{\phi_1, \ldots, \phi_n\}$ yields the $n \times n$ quadratic matrix eigenvalue problem

$$\lambda^2 M x + \lambda\beta F x + (\alpha F + K)x = 0, \tag{6.6}$$

where

$$M_{ij} = \int_\Omega \rho \phi_i \cdot \phi_j, \quad K_{ij} = \int_\Omega \rho c^2 \operatorname{div} \phi_i \operatorname{div} \phi_j, \quad F_{ij} = \int_{\Gamma_A} \phi_i \cdot \nu \phi_j \cdot \nu.$$

To avoid spurious eigenvalues caused by discretization, it is suggested in [6] to use lowest order Raviart–Thomas finite elements [22]. Each basis element ϕ_i is a vector-valued function with piecewise constant divergence on each triangle of the mesh and $\phi_i \cdot \nu$ constant along each edge. With a natural choice of the basis, each finite element corresponds to an edge in the interior or on the absorbing boundary Γ_A. We use a triangulation of Ω with $6N$ edges along the vertical sides and $8N$ edges along the horizontal sides; a model with 9168 degrees of freedom is obtained with the choice of the parameter $N = 8$ (Fig. 2).

Let $A = M$, $B = \beta F$, $C = \alpha F + K$ and write (6.6) as the symmetric quadratic eigenvalue problem

$$(\lambda^2 A + \lambda B + C)x = 0. \tag{6.7}$$

Note that A is symmetric positive definite, and B, C are positive semidefinite matrices. The eigenvalues of interest are those with smaller imaginary parts, corresponding to lower-frequency vibration modes. They have real parts between -250 and -320. Substituting $\widehat{A} = \sigma^2 A + \sigma B + C$, $\widehat{B} = 2\sigma A + B$, $\widehat{C} = A$, and $\mu = (\lambda - \sigma)^{-1}$ gives a shifted and inverted problem

$$\left(\mu^2 \widehat{A} + \mu \widehat{B} + \widehat{C}\right) x = 0;$$

for the choice of $\sigma = -253$, $\widehat{A}$ remains positive definite. Therefore, we can take the Cholesky decomposition $\widehat{A} = LL^{\mathrm{T}}$ and construct the equivalent monic problem

$$\left(\mu^2 I + \mu(L^{-1}\widehat{B}L^{-\mathrm{T}}) + (L^{-1}\widehat{C}L^{-\mathrm{T}})\right)u = 0. \tag{6.8}$$

Algorithm 3.1 is now applied to (6.8) in a symmetric Lanczos type process to get a basis V_k and banded $k \times k$ matrices T_a, T_b. It follows that if (θ_i, u_i) is an eigenpair to the projected problem

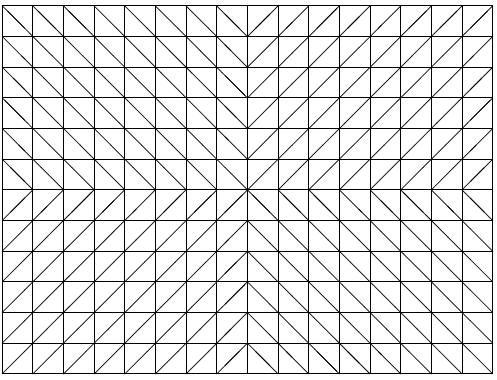


Fig. 2. Triangulation of Ω with $N = 2$.

$$\left(\theta^2 I_k + \theta T_{a(1:k,1:k)} + T_{b(1:k,1:k)}\right)u = 0, \tag{6.9}$$

then $(\lambda_i, x_i) = (\sigma + 1/\theta_i, z_i/\|z_i\|)$ is an approximate eigenpair to the original problem (6.7), where $z_i = L^{-\mathrm{T}} V_{k(:,1:i)} u_i$.

We applied Algorithm 3.1 to compute two conjugate pairs and two real eigenvalues of (6.7). As both the full-scale eigenvalue problem and its projection are real symmetric, all of their eigenvalues are real or appear in conjugate pairs. Thus, both eigenvalues of (6.7) in a conjugate pair converge simultaneously. For each selected eigenvalue and each iteration $i = 1, \ldots, 500$, we compute the corresponding approximate eigenpair (λ_i, x_i) by linearizing and solving (6.9) with the MATLAB `eigs` function. The resulting relative residual norms $r_i = \frac{\|(\lambda_i^2 A + \lambda_i B + C)x_i\|}{|\lambda_i|^2\|A\| + |\lambda_i|\|B\| + \|C\|}$ are plotted in Fig. 3.

The following table lists the values of the four selected eigenvalues. For each eigenvalue $\hat{\lambda}_i$ (as computed by MATLAB), the table gives the type of line used in Fig. 3 to plot the corresponding residual norms, the number of matrix–vector products required to obtain a relative residual norm $r_i < 10^{-8}$, and the error of the corresponding eigenvalue λ_i at that point.

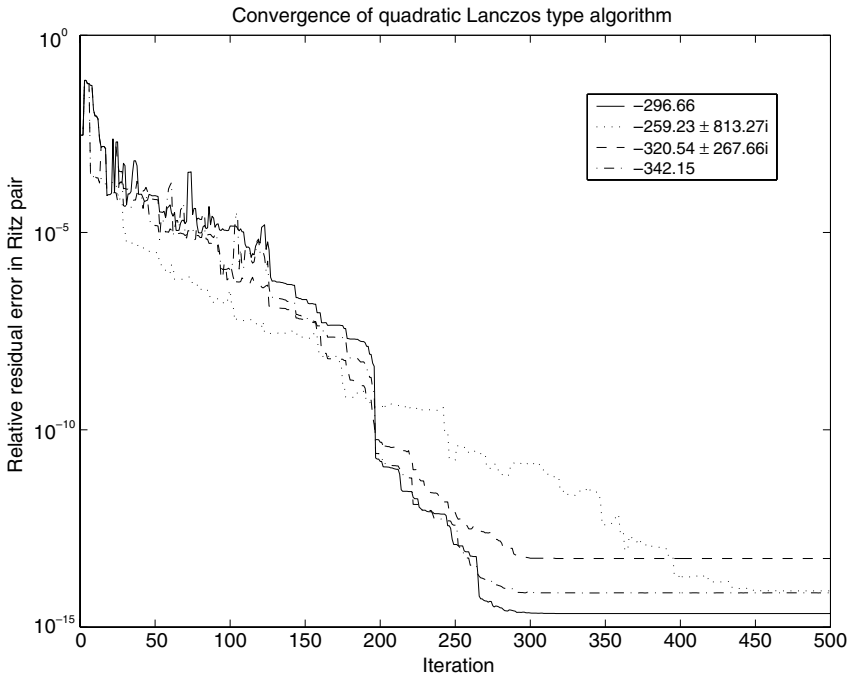


Fig. 3. Relative residual norms for selected eigenvalues.

Eigenvalue λ_i	Plot line	Matrix–vector products	$\lvert\lambda_i - \hat{\lambda}_i\rvert$
$-259.23 \pm 813.27\,\mathrm{i}$	Dotted	318	1.7×10^{-8}
$-320.54 \pm 267.66\,\mathrm{i}$	Dashed	322	1.0×10^{-8}
-342.15	Dash-dot	356	8.8×10^{-9}
-296.66	Solid	386	3.8×10^{-9}

7. Conclusions

We have presented basic properties for some Arnoldi and Lanczos type processes for a monic matrix polynomial with large and sparse coefficient matrices. These processes operate on the same space as the matrix polynomial live and the reduced problem are a matrix polynomial itself. This has the advantage of preserving certain properties of the original system in the reduced systems and the process could hold the key on practical applications where such a feature is a necessity.

What we have presented here are basic ideas. Robust implementations of these new algorithms will require carefully dealing with many (subtle) technical details. Bearing similarity in nature to the standard Arnoldi and Lanczos processes, these new Arnoldi and Lanczos type processes could incorporate many proven techniques developed over the years for the former. We shall leave these matters to future investigations.

Acknowledgements

This work started as search for an answer to a question posed to us by Professor Zhaojun Bai of the University of California at Davis. The question was to produce a physically meaningful reduced order model when the original model is defined by a quadratic matrix polynomial. We are indebted to Prof. Bai for the question. We thank the referees for their careful reading of the manuscript and their constructive comments that have improved the presentation.

References

[1] W.E. Arnoldi, The principle of minimized iterations in the solution of the matrix eigenvalue problem, Quart. Appl. Math. 9 (1951) 17–29.
[2] Z. Bai, Personal communication, 2000.
[3] Z. Bai, D. Day, Q. Ye, ABLE: an adaptive block Lanczos method for non-Hermitian eigenvalue problems, SIAM J. Matrix Anal. Appl. 20 (1999) 1060–1082.
[4] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, H. van der Vorst (Eds.), Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide, SIAM, Philadelphia, 2000.
[5] Z. Bai, Q. Ye, Error estimation of the Padé approximation of transfer functions via the Lanczos process, Electron. Trans. Numer. Anal. 7 (1998) 1–17.

[6] A. Bermúdez, R.G. Durán, R. Rodríguez, J. Solomin, Finite element analysis of a quadratic eigenvalue problem arising in dissipative acoustics, SIAM J. Numer. Anal. 38 (2000) 267–291.
[7] J. Demmel, Applied Numerical Linear Algebra, SIAM, Philadelphia, 1997.
[8] R.J. Duffin, The Rayleigh–Ritz method for dissipative or gyroscopic systems, Quart. Appl. Math. 18 (1960) 215–221.
[9] T. Ericsson, A. Ruhe, The spectral transformation Lanczos method for the numerical solution of large sparse generalized symmetric eigenvalue problems, Math. Comp. 35 (1980) 1251–1268.
[10] P. Feldman, R.W. Freund, Efficient linear circuit analysis by Padé approximation via the Lanczos process, IEEE Trans. Computer-Aided Design 14 (1995) 639–649.
[11] K. Gallivan, E. Grimme, P. van Dooren, Asymptotic waveform evaluation via a Lanczos method, Appl. Math. Lett. 7 (5) (1994) 75–80.
[12] I. Gohberg, P. Lancaster, L. Rodman, Matrix Polynomials, Academic Press, New York, 1982.
[13] G.H. Golub, C.F. van Loan, Matrix Computations, third ed., Johns Hopkins University Press, Baltimore, MD, 1996.
[14] M.H. Gutknecht, A complete theory of the unsymmetric Lanczos process and related algorithms: Part I, SIAM J. Matrix Anal. Appl. 13 (1992) 594–639.
[15] M.H. Gutknecht, A complete theory of the unsymmetric Lanczos process and related algorithms: Part II, SIAM J. Matrix Anal. Appl. 15 (1994) 15–58.
[16] L. Hoffnung, R.-C. Li, Q. Ye, Krylov type subspace methods for matrix polynomials, Technical Report 2002-08, Department of Mathematics, University of Kentucky, 2002. Available from: <http://www.ms.uky.edu/∼math/MAreport/>.
[17] C. Lanczos, An iteration method for the solution of the eigenvalue problem of linear differential and integral operators, J. Res. of the National Bureau of Standards 45 (1950) 255–282.
[18] R.-C. Li, Q. Ye, A Krylov subspace method for quadratic matrix polynomials with application to constrained least squares problems, SIAM J. Matrix Anal. Appl. 25 (2) (2003) 405–428.
[19] K. Meerbergen, Locking and restarting quadratic eigenvalue solvers, SIAM J. Sci. Comput. 22 (2001) 1814–1839.
[20] B.N. Parlett, The Symmetric Eigenvalue Problem, SIAM, Philadelphia, 1998, this SIAM edition is an unabridged, corrected reproduction of the work first published by Prentice-Hall, Inc., Englewood Cliffs, NJ, 1980.
[21] B.N. Parlett, D.R. Taylor, Z.A. Liu, A look-ahead Lanczos algorithm for unsymmetric matrices, Math. Comp. 44 (1985) 105–124.
[22] P.A. Raviart, J.M. Thomas, A mixed finite element method for second order elliptic problems, in: Mathematical Aspects of Finite Element Methods, Lecture Notes in Math., vol. 606, Springer-Verlag, Berlin, 1977, pp. 292–315.
[23] A. Ruhe, Rational Krylov sequence methods for eigenvalue computation, Linear Algebra Appl. 58 (1984) 391–405.
[24] Y. Saad, Numerical Methods for Large Eigenvalue Problems, Manchester University Press, Manchester, UK, 1992.
[25] Y. Saad, Iterative Methods for Sparse Linear Systems, PWS Publishing Company, Boston, MA, 1996.
[26] G. Sleijpen, J. Booten, D. Fokkema, H. van der Vorst, Jacobi–Davidson type methods for generalized eigenproblems and polynomial eigenproblems, BIT 36 (1996) 593–633.
[27] D.C. Sorensen, Implicit application of polynomial filters in a k-step Arnoldi method, SIAM J. Matrix Anal. Appl. 13 (1) (1992) 357–385.
[28] F. Tisseur, K. Meerbergen, The quadratic eigenvalue problem, SIAM Rev. 43 (2) (2001) 235–386.
[29] C.E. Villemagne, R.E. Skelton, Model reduction using a projection formulation, Internat. J. Control 46 (6) (1987) 2141–2169.
[30] Q. Ye, A convergence analysis for nonsymmetric Lanczos algorithms, Math. Comp. 56 (1991) 677–691.
[31] Q. Ye, A breakdown-free variation of the nonsymmetric Lanczos algorithms, Math. Comp. 62 (1994) 179–207.